Theoretical Analysis of Expected Payoff in an Adversarial Domain

Pablo J. Villacorta, David A. Pelta

Models of Decision and Optimization Research Group
CITIC-UGR, Dept. of Computer Science and A.I.
University of Granada, 18071 - Granada, Spain

Abstract

Adversarial decision making is aimed at finding strategies for dealing with an adversary who observes our decisions and tries to learn our behaviour pattern. Based on a simple mathematical model, the present contribution provides analytical expressions for the expected payoff when using simple strategies which try to balance confusion and payoff. Additional insights are provided regarding the structure of the payoff matrix. Computational experiments show the agreement between theoretical expressions and empirical simulations, thus paving the way to make the assessment of new strategies easier.

Key words: adversarial reasoning, decision making, strategies

1 Introduction

Adversarial decision making is aimed at determining successful strategies against an adversarial enemy who observes our actions and learns from them. This situation arises in many areas of real life, with particular (but not exclusive) interest in counter-terrorist combat and crime prevention [4,8].

The field is also known as decision making in the presence of adversaries and we can talk about problems within an adversarial domain where an "adversary" exists. Nowadays there is a quest for technologies aiming at opponent strategy prediction; plan recognition; deception discovery and planning; among others, and which apply not only to security issues but also to the games industry, to business, to transactions, etc. [3]. For example, patrolling strategies can be viewed as another

Email address: `olbapjo@correo.ugr.es, dpelta@decsai.ugr.es` (Pablo J. Villacorta, David A. Pelta).

application of adversarial decision making: the aim is to design routes for mobile robots to patrol which minimize the chances of an enemy entering a security border. References [1,2,5] provide good examples on this topic.

Pelta and Yager [7] proposed a simple adversarial model where two agents or entities S and T (the adversary) seek to maximize their rewards, but which are inversely related. The agents engage in a sort of "imitation game" where given an input, they should issue a response. Agent T wants to mimic the responses of S (acting as an imitator), while agent S tries to avoid being imitated. One defense for S is to make responses that are intended to confuse T, although this will affect the ability of obtaining a greater reward. The focus is on designing decision or response strategies for S that minimize the losses, which are either due to correct guesses or to non-optimal responses.

In such work, some decision strategies for both agents were proposed and empirically evaluated using stochastic simulations. Here, our global aim is to analyze such strategies from a theoretical point of view so that they can be evaluated without running such simulations. In turn, this will lead to a faster and more exact way of comparing strategies, it will facilitate comparisons with new strategies investigated in further research and will allow to understand the impact of certain components of the model. Some other recent work has also been done on this model, although it followed a heuristic, non-exact approach to automatic design of strategies [6,9].

With this in mind, the objectives of this work are: (a) to provide analytical expressions for the expected payoff, based in concepts of probability theory; (b) to show the agreement between the theoretical expected payoff and the empirical results; and (c) to discuss how the strategy used and the definition of the payoff matrix affect the results.

This contribution is organized as follows. In section 2, the model explained in [7] is briefly summarized. Section 3 presents an explanation of the details and assumptions needed to obtain the expressions of the expected payoff (with and without adversary) for each strategy, both in a general case and also in the particular conditions in which our experiments have been conducted. Section 4 provides graphical representations of the results in order to contrast the expected and empirical payoff, and also explains the behaviour of each strategy in terms of the payoff matrices employed. Finally, section 5 discusses the benefits of the results and provides new research lines in this direction.

2 Adversarial Model

The model presented in [7] is based on two agents S and T (the adversary), a set of possible inputs or events $E = \{e_1, e_2, \ldots, e_n\}$ issued by a third agent R, and a

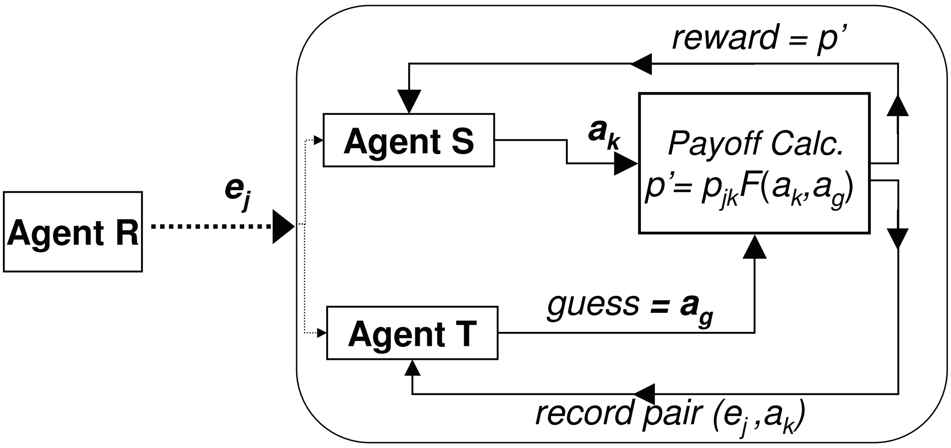

Fig. 1. Graphical representation of the model. Events e_j are issued by agent R while the responses or actions a_k are taken by agent S.

set of potential responses or actions $A_i = \{a_1, a_2, \ldots, a_m\}$ associated with every event. We have a payoff or rewards matrix P :

$$P(n \times m) = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ p_{31} & p_{32} & \cdots & p_{3m} \\ & & & \\ p_{n1} & p_{n2} & \cdots & p_{nm} \end{pmatrix}$$

where $p_{ij} \in [0, 1]$ is the payoff associated with action a_j in response to event e_i.

Agent S must decide which action to take, given a particular input e_j and with a perfect knowledge of the payoff function P. His aim is to maximize the sum of the profits or rewards for a given sequence of inputs. These are issued one at a time and they come from an external environment, represented by agent R. For the experiments, the inputs of the sequence are independent and generated randomly. A scheme of the model is shown in Fig. 1.

Agent T does not know the payoff function P but is watching agent S in order to learn from his actions. His aim is to reduce agent's S payoff by guessing which action he will take as a response to each input of the sequence in a sort of imitation game. Algorithm 1 describes the steps of the model, with L being the length of the sequence of inputs.

Given a new input e_j, S and T issue responses a_k and a_g respectively. At the moment of the response, neither of the agents knows which action the other agent will choose. The payoff for S is computed as a function of both responses and the payoff p_jk. After the payoff has been calculated, agent T is informed of what S had

Algorithm 1 Sequence of steps in the model.

```
for $l = 1$ to $L$ do
    A new input $e_j$ arises.
    Agent $T$ "guesses" an action $a_g$
    Agent $S$ determines an action $a_k$
    Calculate payoff for $S$ as a function of $p_{jk}, a_g, a_k$
    Agent $T$ records the pair $e_j, a_k$
end for
```

chosen, and then T "records" the pair (e_j, a_k) in his own memory. This information can be used in the future by T to make his predictions. The memory in which agent T keeps records of the actions taken by S is modeled as an observation matrix O, with dimensions $N \times M$. O_{ij} stores the number of times that, in the past, agent S took action a_j when the input was e_i.

The reward calculation for S at stage l is defined as:

$$p' = p_{jk} \;\; F(a_g, a_k) \tag{1}$$

where F is:

$$F(a,b) = \begin{cases} 0 & \text{if a = b} \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

This means that agent S obtains no reward when agent T matches his response.

2.1 Behaviour of the agents

In [7], the authors presented and evaluated several strategies for agent S and T. From the point of view of agent T, a good strategy is the so-called *proportional to frequency*. It means that the probability of selecting an action as a prediction for a given event is proportional to the number of times (as a percentage) that that action has been observed in the past as a response to the same stimulus (this information is stored in the observation matrix O). As can be seen, it is also a randomized strategy. In what follows, this will be the strategy assigned for agent T.

From the point of view of S and given an observed input e_i and the payoff matrix P, the following strategies will be considered:

- **Random among k Best actions (R-k-B)**: Randomly select an action from the k best ones.

- **Proportionally Random (PR)**: The probability of selecting an action is proportional to its payoff.
- **Best of k selected Randomly (B-k-R)**: Select the best action (in terms of payoff) from a set of k possible actions selected randomly.

The strategy *Best action* will not be evaluated theoretically because it was proved in [7] to be the worst one. The strategy *Random* (completely random) is a particular case of R-k-B in which k equals the total number of possible actions.

3 Calculation of the Expected Payoff

In this section, we will provide analytical expressions to calculate the expected payoff for S when using the the previous strategies. The basic assumptions are: agent T uses the *proportional to frequency* strategy, and the payoff matrix has a specific structure, which is defined below in the following section.

3.1 Notation and payoff matrix definition

Due to the need to include a certain amount of confusion (randomness) in the strategies for adversarial domains, the payoff matrix plays a key role in the calculation of the expected payoff.

Intuitively, when all the actions have quite similar payoffs, then it is not very important if a sub-optimal action is chosen for a given input. However, it becomes much more problematic if the actions' payoffs are very different, because all the sub-optimal actions provide payoffs that are considerably lower than the best one, thus leading to serious losses.

In this work, we will define the payoff matrices in terms of a parameter α standing for the gap between the best payoff and the rest. All matrices tested are square (the same number of stimuli and responses, namely K_{max}). Let p_{ij} be the elements of the payoff matrix P described in section 2. Every row of P is a permutation of the payoffs in the set Y = $\{$1, 1-α, 1-2α, ... 1-(K_{max}-1)$\alpha\}$. The repetition of the same value is not allowed within a row of the matrix, in order to simplify the mathematical expressions. We will refer to these values (not necessarily in this order) as $r_i, i = 1, \ldots, K_{max}$. Of course $r_i > 0$, $\forall$ i. Let r^i be the i-th greatest value of Y. Under these considerations, matrices generated with lower values of α will have more similar values in each row, whereas higher values of α lead to very different payoffs in the same row.

Finally, let X be the random variable associated with the occurrence of every stimulus, and let $P(X = e_t)$ be the (discrete) probability distribution of the stimuli.

3.2 Expected payoff of R-k-B

The strategy *Random among k best actions* chooses one action (with uniform probability) from a set composed by the k best actions for the current input, being k a parameter of the strategy ($k \leq K_{max}$). Each candidate action can be selected with probability $1/k$, while any action that is not among the k best actions will never be chosen. Let $p_t^1, \ldots, p_t^{K_{max}}$ be the values of row t of the payoff matrix, sorted in descending order (i.e. $p_t^i > p_t^j$ when $i < j$). This is just a notation convention to indicate which the highest payoffs in a row are. Then, when there is no adversary, the total payoff E for agent S after a sequence of L inputs can be expressed as a function of the k value used in the ***R-k-B*** strategy, as follows:

$$E_{RkB}(k) = L \sum_{t=1}^{K_{max}} \left(P(X = e_t) \sum_{i=1}^{k} \frac{1}{k} p_t^i \right) \tag{3}$$

We have considered two different random events in this formula. First, the random variable T, associated to the probability that a given stimulus arises. In the original model, this distribution was considered uniform, so $P(X = e_t) = 1/K_{max}\ \forall t$ when the problem instance has K_{max} different stimuli. However, the above expression was generalized to consider any probability distribution for the stimuli of the sequence. Second, after a stimulus arises, any of the k best actions can be chosen with the same probability, 1/k. As both events are independent, the probability that they occur simultaneously is the product of the probabilities. So, we have:

$$\begin{aligned} E_{RkB}(k) &= L \sum_{t=1}^{K_{max}} \left(\frac{1}{K_{max}} \sum_{i=1}^{k} \frac{1}{k} p_t^i \right) \\ &= \frac{L}{K_{max}\ k} \sum_{t=1}^{K_{max}} \sum_{i=1}^{k} p_t^i \end{aligned} \tag{4}$$

Due to the way we defined the payoff matrices, the set of values in every row is the same, so after sorting, p_t^i are the same regardless of the row t. As a consequence we have

$$\sum_{t=1}^{K_{max}} \sum_{i=1}^{k} p_t^i = K_{max} \sum_{i=1}^{k} r^i \tag{5}$$

for any $k \in [1, K_{max}]$, with r^i being the values of any row of the matrix, sorted in decreasing order.

Every row in the matrix has the values $1, 1-\alpha, 1-2\alpha, \ldots, 1-(K_{max}-1)\alpha$, so the following expression can be proven by mathematical induction:

$$\sum_{i=1}^{k} r^i = \sum_{i=1}^{k} (1 - (i-1)\alpha) = k - \alpha \left(\frac{k^2 - k}{2} \right) \tag{6}$$

Considering these simplifications, the expression in Eq. 3 yields

$$E_{RkB}(k) = \frac{L}{k}\left(k - \alpha\left(\frac{k^2 - k}{2}\right)\right) \tag{7}$$

which represents the expected payoff for agent S without adversary.

When dealing with an adversary we also have to consider the probability of not being guessed in order to obtain a non zero reward for that action. The *R-k-B* strategy implies that, theoretically, after a certain number of inputs, any of the k best actions has been chosen the same number of times and the rest have never been chosen. Thus, we can suppose that the observed frequency for each of the k best actions is the same after a long-enough input sequence, so the probability that agent T, who is using a strategy proportional to frequency, guesses one of them is also $1/k$. Hence the probability of not being guessed is $(k-1)/k$. The probability of simultaneously choosing an action and not being guessed can be calculated as the product of the probabilities of both events since they are independent. Thus it is enough to include the factor $(k-1)/k$ (probability of not being guessed) in the general expression of Eq. 3, obtaining:

$$\begin{aligned} E_{RkB}(k) &= L\sum_{t=1}^{K_{max}}\left(P(X = e_t)\sum_{i=1}^{k}\frac{1}{k}\frac{k-1}{k}p_i^t\right) \\ &= L\frac{k-1}{k^2}\sum_{t=1}^{K_{max}}\left(P(X = e_t)\sum_{i=1}^{k}p_i^t\right) \end{aligned} \tag{8}$$

Finally, considering simplifications due to a uniform distribution for the stimuli and the particular form of our payoff matrices (recall 5 and 6), we have:

$$E_{RkB}(k) = L\frac{k-1}{k^2}\left(k - \alpha\left(\frac{k^2 - k}{2}\right)\right) \tag{9}$$

which represents the expected payoff for agent S when using the *Random among k best actions* in the presence of an adversary.

3.3 Expected payoff of Proportionally Random

In this strategy (henceforth called PR), the probability of choosing an action is proportional to its payoff. Let z_{ti} be the probability of agent S choosing action i as a response to stimulus t using strategy PR. By definition of PR we have

$$z_{ti} = \frac{p_{ti}}{\sum_{k=1}^{K_{max}} p_{tk}} \tag{10}$$

The total expected payoff, thus, can be calculated as the sum of the expected payoff for all the possible actions. This idea yields the following general expression of the expected payoff E_{PR} after a sequence of L inputs when there is no adversary.

$$\begin{aligned} E_{PR} &= L \sum_{t=1}^{K_{max}} \left(P(X = e_t) \sum_{i=1}^{K_{max}} z_{ti} p_{ti} \right) \qquad (11) \\ &= L \sum_{t=1}^{K_{max}} \left(P(X = e_t) \sum_{i=1}^{K_{max}} \frac{p_{ti}}{\sum_{k=1}^{K_{max}} p_{tk}} p_{ti} \right) \qquad (12) \end{aligned}$$

Notice that this strategy does not require any additional parameter like the preceding *R-k-B* strategy.

Now let us consider again the definition of z_{ti}. Theoretically, after a certain number of inputs, say L (suppose L is *long enough*), action i will have been chosen z_{ti} L times. As the adversary agent T always uses a strategy proportional to the observed frequency to make his prediction, the probability that T guesses that action correctly can be calculated as the number of times he has observed that action when the input was t, divided by the total number of observations, L. In other words, the probability that T chooses action i is

$$\frac{z_{ti} L}{L} = z_{ti}$$

Thus, the probability that T does not guess correctly is 1-z_{ti}.

Finally, the expected payoff for action i when dealing with an adversary can be calculated as the basic payoff of action i, p_{ti}, weighted by the probability of simultaneously choosing action i, z_{ti} and by the probability of not being predicted properly, 1-z_{ti}. Once again, both events are independent so the probability that they happen simultaneously is the product of the probabilities of each individual event. In other words, we just have to incorporate to (11) the factor 1-z_{ti}, which yields

$$\begin{aligned} E_{PR} &= L \sum_{t=1}^{K_{max}} \left(P(X = e_t) \sum_{i=1}^{K_{max}} z_{ti}(1 - z_{ti}) p_{ti} \right) \\ &= L \sum_{t=1}^{K_{max}} \left(P(X = e_t) \sum_{i=1}^{K_{max}} \frac{p_{ti}}{\sum_{k=1}^{K_{max}} p_{tk}} \left(1 - \frac{p_{ti}}{\sum_{k=1}^{K_{max}} p_{tk}} \right) p_{ti} \right) \qquad (13) \end{aligned}$$

Again, we can simplify this expression due to a uniform distribution for the stimuli and the particular form of our payoff matrices. Eq. 6 still holds and the whole inner summation is constant regardless of the row t because all the rows have a permutation of the same set of payoffs. This yields

$$\begin{aligned} E_{PR} &= L\frac{1}{K_{max}}\sum_{t=1}^{K_{max}}\left(\sum_{i=1}^{K_{max}}\frac{p_{ti}}{\sum_{k=1}^{K_{max}}p_{tk}}\left(1-\frac{p_{ti}}{\sum_{k=1}^{K_{max}}p_{tk}}\right)p_{ti}\right) \\ &= L\frac{1}{K_{max}}K_{max}\left(\sum_{i=1}^{K_{max}}\frac{r_i}{\sum_{k=1}^{K_{max}}r_k}\left(1-\frac{r_i}{\sum_{k=1}^{K_{max}}r_k}\right)r_i\right) \\ &= L\left(\sum_{i=1}^{K_{max}}\frac{r_i}{C}\left(1-\frac{r_i}{C}\right)r_i\right) \end{aligned} \tag{14}$$

with r_i being the values of any row of the payoff matrix but now in no specific order. The sum of payoffs C is defined as: $C = K_{max} - \alpha(\frac{K_{max}^2 - K_{max}}{2})$

3.4 Expected payoff of B-k-R

This strategy first randomly chooses k different (*candidate*) actions, and secondly, the best of such actions is finally selected as a response to a given stimulus. Here, k is a parameter of the strategy.

Let a_i with $i = 1, \ldots, K_{max}$ be one of the candidate actions agent S can choose to answer to stimulus t, and let p_{ti} be the corresponding payoff. Action a_i will finally be chosen if, and only if, the following two conditions are met:

(1) action a_i must be chosen as one of the k candidate actions
(2) the candidate set must not contain any other action whose payoff is greater than p_{ti}, because if that were the case, then action a_i would not be selected as the *Best-among-k* candidate actions.

Any action will appear in the set of candidate actions with probability k/K_{max}. When a certain action a_i has been chosen for the candidate set, it will be finally selected as the actual response if, and only if, there are no better actions in the candidate set. The probability that this occurs can be obtained as the quotient of favorable cases divided by the total number of feasible cases.

The number of feasible cases is the number of non-sorted combinations of $k - 1$ actions (the rest of the actions, apart from action a_i that is being studied) taken from the total set of remaining actions, which has $K_{max} - 1$ elements because action a_i has already been picked. The mathematical concept of *combination* captures this notion. By definition, the number of b-combinations (each of size b) from a set with

a elements (size a) is the binomial coefficient:

$$comb(a,b) = \binom{a}{b} = \frac{a!}{b!(a-b)!}$$

so the number of feasible cases we need is $\binom{K_{max}-1}{k-1}$.

The number of cases that are favorable to action a_i can be computed as follows. A set of k candidate actions is favorable to action a_i if all the actions in the set have lower payoffs than a_i. The number of favorable cases is the number of favorable combinations, i.e. combinations of (k-1) actions taken only from the subset of actions that are worse than a_i.

The restriction of using only actions that are worse than a_i is the key to calculating the number of favorable combinations. The number of actions that have a better payoff than p_{ti} can be expressed as a function $B : \mathbb{R} \rightarrow \mathbb{N}$ which takes payoffs into naturals, so the number of actions that are worse can be calculated as $K_{max} - 1 - B(p_{ti})$. As we will show, we use this function to make the next expressions valid for any payoff matrix (although our particular matrices lead to simpler expressions). So, the number of favorable combinations is $\binom{K_{max}-1-B(p_{ti})}{k-1}$.

Once we have calculated the probability of each action being chosen, we can use this to obtain the general expression of the expected payoff of ***B-k-R*** for a sequence of L inputs when there is no adversary:

$$E_{BkR}(k) = L \sum_{t=1}^{K_{max}} \left(P(X = e_t) \sum_{i=1}^{K_{max}} \frac{k}{K_{max}} \cdot \frac{\binom{K_{max}-1-B(p_{ti})}{k-1}}{\binom{K_{max}-1}{k-1}} p_{ti} \right) \quad (15)$$

Again, we can simplify the above formula as has been done in the previous cases. If the stimulus follows a uniform probability distribution, then $P(X = e_t) = 1/K_{max}\ \forall t \in \{1, \ldots, K_{max}\}$. Also, if the payoff matrix has a permutation of the same set of values in each row, as in our particular payoff matrices, then the result of the inner summation is constant regardless of the value of t, so the outer summation can be ignored whilst the inner can be substituted by K_{max} multiplied by the same inner summation but now ignoring index t and using r_i. These are the same transformations we have already carried out in the preceding sections.

$$\begin{aligned} E_{BkR}(k) &= L \frac{K_{max}}{K_{max}} \sum_{i=1}^{K_{max}} \frac{k}{K_{max}} \frac{\binom{K_{max}-1-B(r_i)}{k-1}}{\binom{K_{max}-1}{k-1}} r_i \\ &= L \frac{k}{K_{max}} \sum_{i=1}^{K_{max}} \frac{\binom{K_{max}-1-B(r_i)}{k-1}}{\binom{K_{max}-1}{k-1}} r_i \end{aligned} \quad (16)$$

Two more simplifications are possible for our particular matrices. First, we can suppose again that the payoffs are sorted in decreasing order (from best to worst), so $r_i > r_j$ when $i < j$. This supposition does not change the result of the summation; it just re-sorts its terms. Using this fact and considering that payoff values are not repeated within a row, function B becomes trivial: $B(r_i) = i - 1$, with $i = 1, \ldots, K_{max}$. The reader should note that r_1 is the payoff of the best action, so there are 0 actions better than it, and $r_{K_{max}}$ is the worst action because all the $K_{max} - 1$ remaining actions are better. Also, as every row is a permutation of $\{1, 1-\alpha, \ldots, 1-(K_{max}-1)\alpha\}$, then $r_i = 1 - (i-1)\alpha$. This leads to the final expression of the expected payoff without an adversary:

$$E_{BkR}(k) = L\frac{k}{K_{max}} \sum_{i=1}^{K_{max}} \frac{\binom{K_{max}-1-(i-1)}{k-1}}{\binom{K_{max}-1}{k-1}}(1-(i-1)\alpha) \tag{17}$$

Finally, we can apply the same reasoning used in the previous section in order to calculate the probability of not being guessed. If z_{ti} represents the probability that agent S chooses an action, then it also represents the probability of being guessed when T uses a strategy proportional to the observed frequency. So we need to introduce the factor $1 - z_{ti}$ into the expressions above. It should be recalled that, in strategy ***B-k-R***, we have

$$z_{ti} = \frac{k}{K_{max}} \frac{\binom{K_{max}-1-B(p_{ti})}{k-1}}{\binom{K_{max}-1}{k-1}}$$

so expression 15 turns into the following expression in order to include this probability of not being guessed:

$$E_{BkR}(k) = L \sum_{t=1}^{K_{max}} \left(P(X = e_t) \sum_{i=1}^{K_{max}} z_{ti}(1 - z_{ti})p_{ti} \right) \tag{18}$$

Using our particular matrix,

$$z'_i = \frac{k}{K_{max}} \cdot \frac{\binom{K_{max}-1-(i-1)}{k-1}}{\binom{K_{max}-1}{k-1}}$$

with z'_i being the probability of choosing action a_i regardless of the row (remembering that every row is a permutation of the same set of values). Then expression 17 becomes:

$$E_{BkR}(k) = L \sum_{i=1}^{K_{max}} z'_i(1 - z'_i)(1 - (i-1)\alpha) \tag{19}$$

which represents the expected payoff for agent S when using strategy ***B-k-R*** in the presence of an adversary.

4 Computational Experiments and Results

In the previous sections we provide analytical expressions for the expected payoff that agent S can achieve using the decision strategies presented above, with and without an adversary.

In this section, we will assess the impact of the payoff matrix on the final payoff achieved by agent S. From a set of matrices, we will calculate the expected payoff using the theoretical expressions and the effective payoff calculated through simulations and then check the agreement between both results.

We will test several 20x20 matrices (i.e. K_{max} = 20). This means that the values of each row in a given matrix are in the set $\{1, 1-\alpha, \ldots, 1-19\alpha\}$. To avoid negative payoffs, we need $1 - 19\alpha > 0$ so the following inequality holds:

$$1 - 19\alpha \geq 0 \iff \alpha \leq 0.052 \tag{20}$$

A full factorial design of the experiments was performed, where the factors and their levels are the following:

- First factor: $\alpha \in \{0.001, 0.010, 0.020, 0.030, 0.040, 0.050\}$, which means there are six different payoff matrices to be tested with each strategy.
- Second factor: agent S strategy $= \{$***PR***, ***R-k-B***, ***B-k-R***$\}$.
- Third factor: parameter $k \in [2, 3, \ldots, 20]$. This is only considered when using the strategies ***R-k-B*** and ***B-k-R***.

In the third factor, we omitted $k = 1$ for two reasons: first, in ***B-K-R*** it is equivalent to a completely random strategy (which is already included in ***R-K-B*** with k=K_{max}), and second, in ***R-K-B*** it is equivalent to always choosing the best action, which is the worst strategy according to the results presented in [7].

Every combination of α, strategy and k value (if applies) has been evaluated theoretically and empirically. In this last case, we performed 2000 runs/simulations of Algorithm 1 where the length of the input sequences employed was $L = 250$. Inputs were considered independent and they were generated using a uniform distribution. As the best response to any stimulus always returns a payoff of 1, the

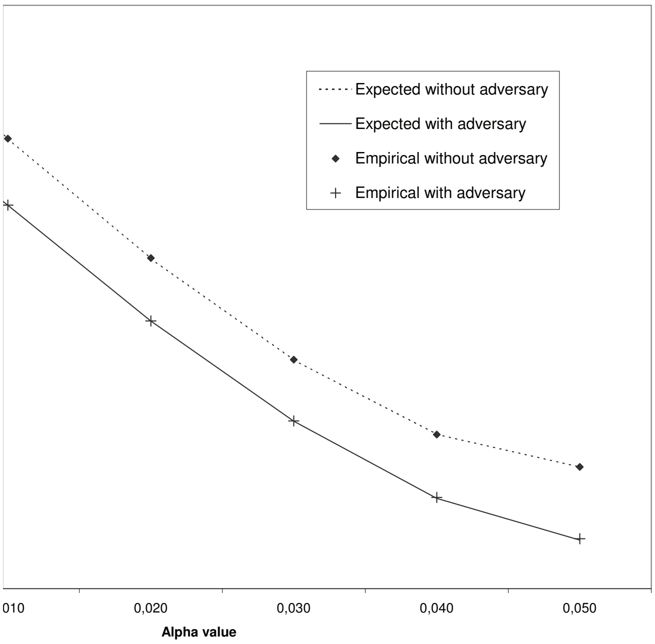

Fig. 2. Expected and empirical payoff for Proportionally Random

maximum possible payoff after a 250-input sequence is precisely 250. This would be the ideal situation. The payoff assigned to the combination is the average payoff over the 2000 simulations.

4.1 Results for Proportionally Random

The empirical and theoretical results regarding the expected payoff as a function of the α value (with and without adversary) are shown in Fig. 2. Empirical results are shown with markers while theoretical ones are displayed with lines. The first element to notice is the perfect agreement between the analytical expressions and empirical results in both cases: with and without adversary.

The relation between α and payoff is the second element to notice. As the former increases, the latter decreases. The reasons for this are the following. As α grows the payoff matrix is less balanced (the differences among payoffs are greater). Using this decision strategy, when a sub-optimal action is selected, the contribution to the total payoff is lower than that which could be achieved selecting the best alternative. In other words, the cost (potential loss in payoff) of choosing a sub-optimal solution is higher. This is clearly observed in Fig. 3 where the payoff assigned to every action is shown. Although the best action always returns a payoff of 1, the other actions' payoffs are diminished as the α value increases.

In turn, as the payoff matrix is less balanced, the probability of choosing one of the best actions grows while the probability of choosing one of the worst actions

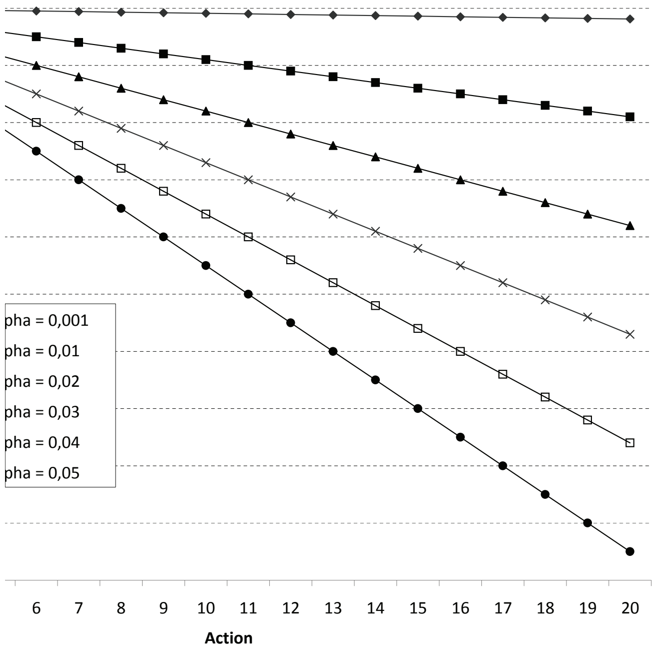

Fig. 3. Payoff per action in every matrix tested

diminishes. Fig. 4 shows this phenomenon. Each line represents the probabilities of choosing each of the twenty possible actions with a different payoff matrix. Here, action 1 is the best while action 20 is the worst. The almost horizontal line corresponds to the matrix generated with α = 0.001 (payoffs and in turn, probabilities are well balanced), while the most-sloped diagonal line corresponds to α = 0.050 (probabilities are very unbalanced). It should be noted how the probabilities of choosing one of the 10 best actions tend to grow while the rest tend to diminish.

Now, the situation is as follows. The best actions have more chances of being selected but, from the point of view of the adversary, they have been more frequently observed. In this situation it would be easier for the adversary to properly predict the action that S will choose.

4.2 Results for R-k-B

In order to analyze R-k-B strategy we would also need to take into account the value of the parameter k.

In Fig. 5 we present the expected and empirical payoff attained using this strategy (without adversary), as a function of k for every payoff matrix tested. Again, we can first observe the perfect agreement between theoretical expressions and empirical results. It is also clear that for a given α, the payoff depends linearly on the value of k, with a higher slope as α increases. As k increases, a highly-randomized strategy is obtained, so the chances of selecting sub-optimal responses also increase.

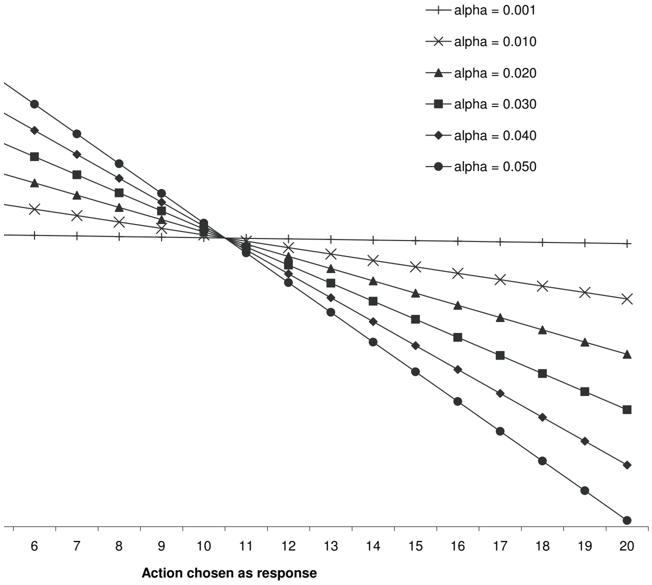

Fig. 4. Probability distribution of the actions for different α values in PR

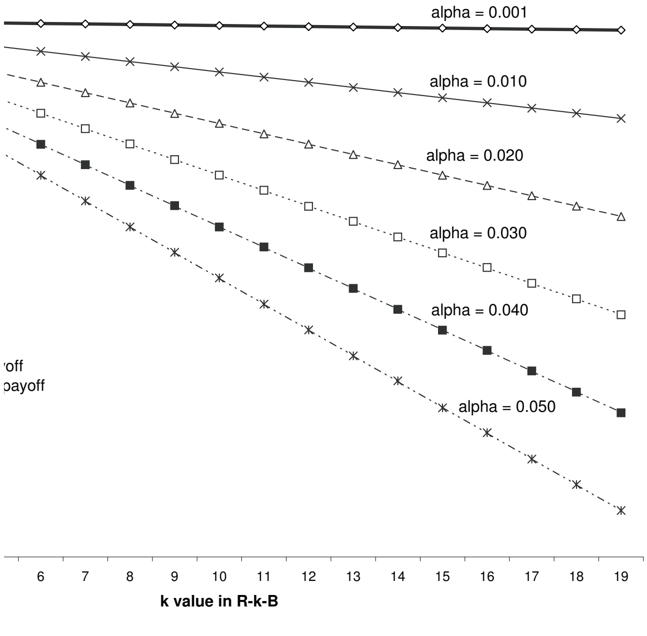

Fig. 5. Expected and empirical payoff for R-k-B without adversary

When α is low, this is not very problematic, but when α increases, the selection of sub-optimal actions with low payoffs leads to substantial losses.

When dealing with an adversary, the situation is very different as can be observed in Fig. 6.

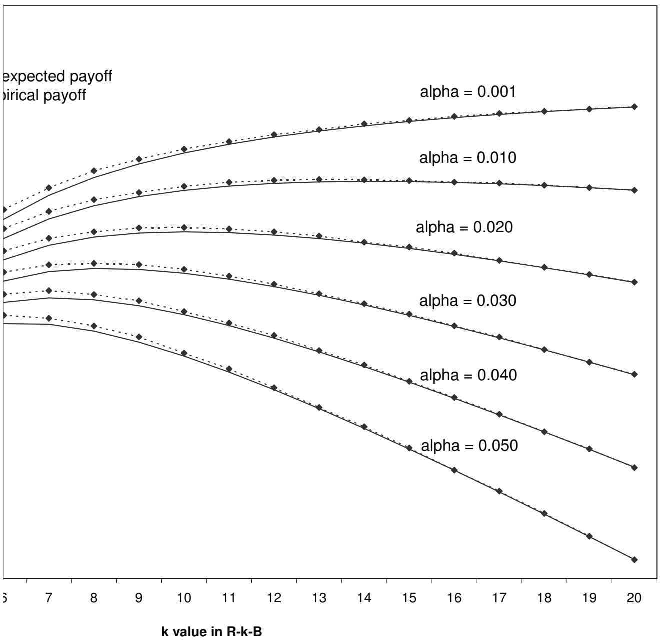

Fig. 6. Expected and empirical payoff for R-k-B with adversary

Again, the agreement between the empirical and theoretical approaches for the payoff calculation is high. In this case, the minor differences between the expected (dashed line) and the empirical (continuous line) payoff at the beginning of each curve can be explained in terms of probabilities. Probability estimations only hold when the experiments are repeated a great number of times [1] .

Here, partially-randomized strategies may be better because they make our behaviour more unpredictable, although sub-optimal actions do not report so much benefit. When the values of the matrix are very similar, increasing k is always better because it results in a more unpredictable behaviour while keeping the total payoff very close to the optimal. When α is high and the values of the matrix are not as similar, a more random behaviour is not always the best alternative because the losses due to sub-optimal actions are important, yet hard to guess. In this situation, the precise amount of randomness needed can be calculated as the exact value for k that maximizes the total payoff for a given payoff matrix. That is the point where the payoff-vs-k curve reaches its absolute maximum and, as can be seen in Fig. 6.

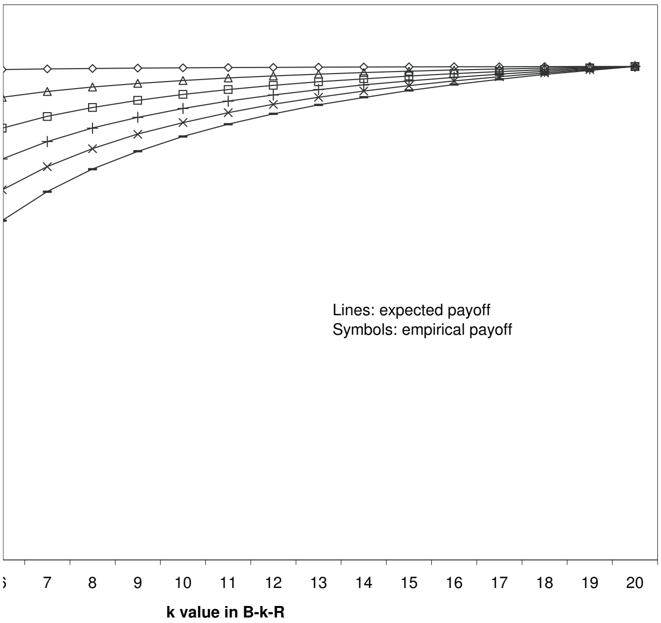

Fig. 7. Expected and empirical payoff for B-k-R without adversary

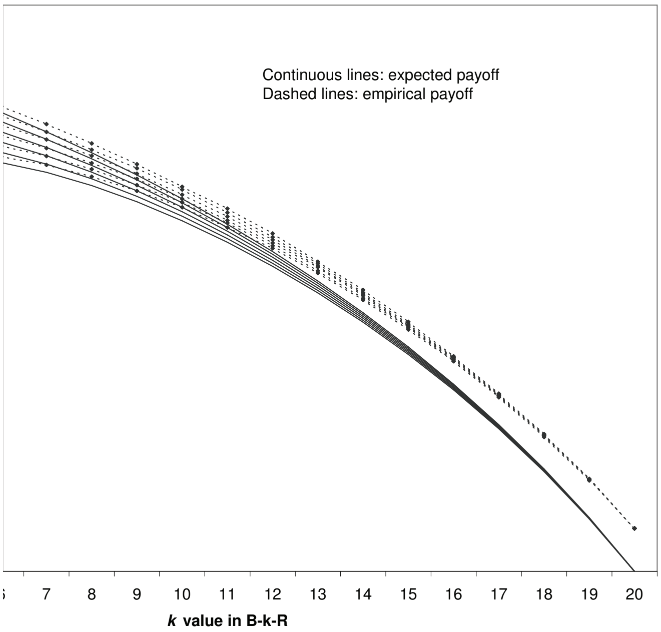

Fig. 8. Expected and empirical payoff for B-k-R with adversary

4.3 Results for B-k-R

In this strategy, we will also consider the results with and without adversary, and also taking into account all the possible values for the parameter k. It should be

[1] We have repeated the experiments with a longer input sequence (1000 stimuli) and these differences become smaller, which means that the formula holds when the input sequence is *large enough*. Results are not shown as they do not substantially contribute to the analysis.

remembered that the strategy randomly selects k actions and then it chooses the best one available.

As expected, when there is no adversary, it is always better to increase the k value because it enlarges the candidate set, which allows to consider more actions and thus to choose the best among them (see Fig. 7). If k equals K_{max}, then all the actions are always selected within the candidate set, thus the strategy degenerates into always selecting the best action. Obviously, this strategy works perfectly only when no adversary is present. On the contrary, if k is low, then the set of candidate actions becomes very randomized, and it may happen quite often that good actions are not selected. When this occurs, losses become more significant, especially if the differences between the payoffs are large (i.e. when α is large). This fact explains the reasons for the different slopes in the curves.

When the adversary is included in the game, the results are those shown in Fig. 8. We can observe that as the value of k increases, the expected payoff is always slightly less than the empirical one. This difference can be easily explained. At the beginning of an experiment, when the observation matrix used by agent T only contains zeros, the first prediction made by T is totally random and possibly wrong. This can happen once per each different input, i.e. up to K_{max} times because the observation matrix has K_{max} rows that are initialized to all 0. As a result, agent S may attain some *extra* payoff at the first steps of each experiment. If the length of the input sequence is large enough, this effect has less influence on the total payoff. However, it can be clearly appreciated in the extreme case, $k = K_{max}$, which is equivalent to always choosing the best action, as explained above. The expected payoff for S in this case is 0 because agent T should always be able to properly guess our action, but the empirical payoff is around K_{max} due to the phenomenon of the initial steps. It should be recalled that the best action has a payoff of 1 and the observation matrix has K_{max} independent rows. The same phenomenon also explains the differences shown in Fig. 6. It is also clear from the plot that as k increases, the impact of the payoff matrix used is diminished.

5 Conclusions and future work

In this paper, we have studied decision strategies in an adversarial model from a theoretical point of view, providing analytical expressions for the expected payoff when the adversary is present or is not. An interesting feature of the model studied is that the assumptions are minimal. We observed an almost perfect agreement between theoretical and empirical results, so the expressions can be used to evaluate the strategies without running a simulation, thus allowing to analyze the impact of model features (such as number of alternatives/stimuli), or to make comparisons with other strategies faster. In addition, we have discussed the role of payoff matrices in the results. Under the particular definition we used, it is clear that such matrices have a great impact in the payoff obtained. More studies are needed using different types of matrices, for example, where the sum of the payoffs in every row is the same, or where different alternatives with the same reward can exist. Game-theoretic studies of equilibrium may also be useful, specially those that make use of particular equilibrium concepts for repeated games.

The understanding of these basic features will allow to develop better strategies for this simple adversarial model (see [10] for more sophisticated dynamic strategies that make use of the same ideas about the expected payoff) and can pave the way to study variations that are yet not considered but realistic, such as adversaries not taking decisions (only observing), or sequence of stimulus with certain correlations (between the last response and the next random stimulus) and so on. Some of these aspects are already under study.

Acknowledgment

This work was supported in part by projects TIN2008-01948 and TIN2008-06872-C04-04 from the Spanish Ministry of Science and Innovation and P07-TIC- 02970 from the Andalusian Government.

References

[1] F. Amigoni, N. Basilico, and N. Gatti. Finding the optimal strategies for robotic patrolling with adversaries in topologically-represented environments. In *Proceedings of the 26th International Conference on Robotics and Automation (ICRA'09)*, pages 819–824, 2009.

[2] F. Amigoni, N. Gatti, and A. Ippedico. A game-theoretic approach to determining efficient patrolling strategies for mobile robots. In *Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology (IAT'08)*, pages 500–503, 2008.

[3] A. Kott and W. M. McEneany. *Adversarial Reasoning: Computational Approaches to Reading the Opponents Mind.* Chapman and Hall/ CRC Boca Raton, 2007.

[4] A. Kott and M. Ownby. Tools for real-time anticipation of enemy actions in tactical ground operations. In *Proceedings of the 10th International Command and Control Research and Technology Symposium*, 2005.

[5] P. Paruchuri, J. P. Pearce, and S. Kraus. Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. In *Proceedings of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS'08)*, pages 895–902, 2008.

[6] D. Pelta and R. Yager. Dynamic vs. static decision strategies in adversarial reasoning. In *Proceedings of the Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference (IFSA-EUSFLAT'09)*, pages 472–477, 2009.

[7] D. Pelta and R. Yager. On the conflict between inducing confusion and attaining payoff in adversarial decision making. *Information Science*, 179:33–40, 2009.

[8] R. Popp and J. Yen. *Emergent Information Technologies and Enabling Policies for Counter-Terrorism*. John Wiley and Sons Hoboken, NJ, 2006.

[9] P. Villacorta and D. Pelta. Evolutionary design and statistical assessment of strategies in an adversarial domain. In *Proceedings of the IEEE Conference on Evolutionary Computation (CEC'10)*, pages 2250–2256, 2010.

[10] P. Villacorta and D. Pelta. Expected payoff analysis of dynamic mixed strategies in an adversarial domain. In *Proceedings of the 2011 IEEE Symposium Series on Computational Intelligence (SSCI'11), Intelligent Agents Conference*, 2011. In press.